

Low-Complexity Cloud Image Privacy Protection via Matrix Perturbation

Xuangou Wu*, Shaojie Tang[†], and Panlong Yang[‡],

*the School of Computer Science and Technology, AHUT, Ma'anshan, China

[†]University of Texas at Dallas, USA

[‡]the Institute of Communication Engineering, PLAUST, Nanjing, China

Emails: wxgou@mail.ustc.edu.cn, tangshaojie@gmail.com, panlongyang@gmail.com

Abstract—Cloud-assisted image services are widely used for various applications. Due to the high computational complexity of existing image encryption technology, it is extremely challenging to provide privacy preserving image services for resource-constrained smart device. In this paper, we propose a novel encrypressive cloud-assisted image service scheme, called eCIS. The key idea of eCIS is to shift the high computational cost to the cloud allowing reduction in complexity of encoder and decoder on resource-constrained device. This is done via compressive sensing (CS) techniques, compared with existing approaches, we are able to achieve privacy protection at no additional transmission cost. In particular, we design an encryption matrix by taking care of image compression and encryption simultaneously. Such that, the goal of our design is to minimize the mutual information of original image and encrypted image. In addition to the theoretical analysis that demonstrates the security properties and complexity of our system, we also conduct extensive experiment to evaluate its performance. The experiment results show that eCIS can effectively protect image privacy and meet the user's adaptive security demand. eCIS reduced the system overheads by up to $4.1\times \sim 6.8\times$ compared with the existing CS based image processing approach.

Index Terms—Privacy-protection, cloud security, compressive sensing, encryption matrix.

I. INTRODUCTION

During the recent years, cloud-assisted image services are widely used for various applications, such as individual image storage and sharing by smartphone (e.g., Facebook and QQ), healthcare image monitoring by wireless sensors [20], [1] and so on. However, this also brings serious security threat to users because of the public access of cloud [2]. Meanwhile, more and more resource-constrained smart devices are used as sensors for sampling these image signals, this also poses a great challenge to the development of appropriate image encryption techniques for mobile device.

Existing cloud security image encryption techniques usually follow compression then encryption paradigm. These techniques require image acquisition device to perform either expensive coding or encryption operation [17], [26] on original image. For example, transformed-based image coding techniques require complex encoding computation. In recent years, compressive sensing (CS) has been proposed for sparse/compressible signal sampling and compression. C. Wang *et.al.* proposed a cloud-assisted computation outsourcing scheme for healthcare video monitoring [25]. They implemented image encryption by linear programming (LP) problem

transformation, which is the application of LP secure outsourcing problem [24]. However, the transformation overhead may easily outweigh the benefits brought by their outsourcing scheme.

As a result, existing results are not appropriate for resource constrained smart device as it requires low-complexity image compression and encryption. In particular, we notice that existing techniques require expensive computation resource mainly due to the following two separated processes: image compression and encryption. To this end, we propose a novel encrypressive Cloud-assisted Image Service scheme (eCIS). In eCIS, we implement image compression and encryption simultaneously via CS, which can significantly reduce the resource consumption for sampling and receiver devices. Meanwhile, we implement cloud storage and computation outsourcing for image user at no additional communication cost. The development of eCIS faces several unique challenges: (1) How to design encryption matrix to meet CS theory and image signal encryption without increasing transmission cost? (2) How to implement encryption matrix to meet low-complexity requirement for encoder and end user? (3) How to guarantee user's privacy performance with our design encryption matrix?

To address the above challenges, we design an encryption matrix based on inverse matrix and mutual information to meet CS encryption encoding and decoding at no additional transmission cost. To achieve low-complexity encryption and decryption for encoder and end user, we exploit adaptive perturbation matrix as our encryption matrix. Moreover, we conduct theoretical analysis that demonstrates several security properties and low-complexity of our scheme. The contributions of this paper are as the follows.

1) We present a novel encrypressive cloud-assisted image service scheme via CS, called eCIS. Our scheme implements the image signal storage and computation outsourcing for both sampling device and end user. Meanwhile, it also can protect user's image signal privacy.

2) We formulate our security problem according to security and system requirements, and design an encryption matrix based on mutual information. Our designed encryption matrix implements compression and encryption simultaneously. We also conduct theoretical analyses of both image security and system overhead.

3) With the extensive experiments, we show that eCIS

not only protect image privacy efficiently but also meet the user's adaptive security demand. The experimental results also display that eCIS decrease $4.1\times \sim 6.8\times$ time cost for sampling device and end user compared with the existing CS based image processing approach.

The rest of this paper is organized as follows. In section 2 presents the related preliminaries. The problem statement is given in section 3. The detailed design of eCIS is presented in Section 4. In section 5, we give the theoretical performance analysis of eCIS. Section 6 reports our experimental results. We present a literature review of existing work in section 7. Finally, we make a conclusion and future work in section 8.

II. PRELIMINARIES

A. CS and Compressive Signal

CS theory asserts that a relatively small number linear combination of a sparse signal could contain most of its salient information [11]. This technique shifts the computation cost from encoder to decoder compared with transformed-based image compression [4]. Assuming that $\mathbf{s} \in \mathbb{R}^n$ is a t -sparse signal, which has only t non-zero components. Thus, the information can be extracted from \mathbf{s} by $\mathbf{y} = \Phi\mathbf{s}$, where Φ is an $m \times n$ measurement matrix, $\mathbf{y} \in \mathbb{R}^m$ is measurement vector and $m \ll n$. If Φ satisfies the restricted isometric property (RIP) and $m \geq O(t \cdot \log(n/t))$ [6], the sparse signal \mathbf{s} could be recovered with high probability. Candès, Romberg, and Tao [5] and Donoho [11] have shown many random matrices that satisfy the RIP such as Gaussian identity distribution matrix, ± 1 Bernoulli matrix and so on. The signal \mathbf{s} could be recovered via ℓ_1 optimization as

$$\hat{\mathbf{s}} = \arg \min_{\mathbf{s}} \|\mathbf{s}\|_1 \quad \text{s.t.} \quad \mathbf{y} = \Phi\mathbf{s}$$

There have been many efficient algorithms to solve the above problems such as basis pursuit [6], orthogonal matching pursuit (OMP) algorithm [23], CoSaMP [18] and so on.

The real image signal, however, is rarely sparse, which can be transformed into sparse signal by a sparse representation basis. In other words, this signal can be well-approximated by a sparse signal, which is called compressible signal. For example, an image signal $\mathbf{x} \in \mathbb{R}^n$ can usually be transformed into a sparse signal \mathbf{s} under discrete cosine transformation (DCT) basis or discrete wavelet transformation (DWT) basis. Given $\mathbf{x} = \Psi\mathbf{s}$ and Ψ is a $n \times n$ representation basis, $\mathbf{s} = [s_1, s_2, \dots, s_n]^T$ is the coefficient vector of \mathbf{x} under Ψ . If \mathbf{x} is compressible, then the magnitudes of the sorted coefficients s_i observe a power-law decay :

$$|s_i| \leq C \cdot i^{-q}$$

where C and $q > 0$ are constants. The compressible signal \mathbf{x} can be represented accurately by only t ($t \ll n$) coefficients [10].

B. Mutual Information

In information theory, mutual information represents the mean relevance of two random variables which is defined as follows.

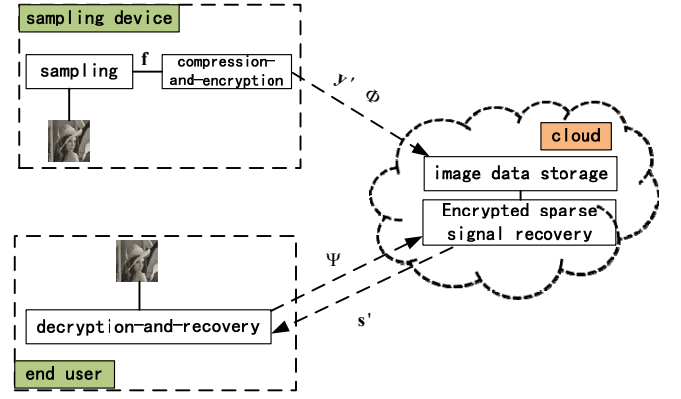


Fig. 1. System architecture of eCIS.

Definition 1. [22] Consider two random variables X and Y with a joint probability mass function $p(xy)$, marginal probability mass functions $p(y)$ and conditional probability function $p(y/x)$. The mutual information $I(X; Y)$ is :

$$\begin{aligned} I(X; Y) &= \sum_{x \in X} \sum_{y \in Y} p(xy) \log_2 \frac{p(y/x)}{p(y)} \\ &= - \sum_{y \in Y} p(y) \log_2 p(y) + \sum_{x \in X} \sum_{y \in Y} p(xy) \log_2 p(y/x) \\ &= H(Y) - H(Y/X) \end{aligned} \quad (1)$$

where $H(Y)$ and $H(Y/X)$ represent entropy and conditional entropy, respectively.

The smaller $I(X; Y)$ is, the less the information of X obtained from Y . If X and Y are two independent random variables, $I(X; Y)$ equals to zero. It means that no information of X could be obtained from Y .

III. PROBLEM STATEMENT

A. System Architecture

Fig.1 shows the overall architecture of eCIS. The system consists of three parties: sampling device, cloud, and end user. The sampling device is responsible for sampling, compressing, and encrypting the original image signal. After image sampling, the device carries out compression and encryption operations. Then the sampling device uploads its compressed and encrypted signal \mathbf{y}' and measurement matrix Φ to the cloud. If an end user wants to access the image, he should send its request and sparse representation basis, Ψ , to the cloud. Cloud receives the user's image request and Ψ , and calculates the sparse signal \mathbf{s}' via CS decoding algorithm and sends it to the end user. After receiving \mathbf{s}' , the end user is responsible for recovering the original image \mathbf{f} by \mathbf{s}' . We assume that the key of encryption matrix cloud be securely transmitted between the sampling device and end user. In practice, sampling device and end user could be the same smart device(e.g., smartphone).

Design goals: Our goals consist of three aspects: (1) The sampling device should implement image compression and

protect the user's image privacy. Meanwhile, the compression and encryption process should have low complexity and transmission cost. (2) The cloud should carry out complicated CS-based image decoding without leaking user's privacy. (3) The image recovery and decryption at end user should also have low complexity.

B. CS-based Image Compression

Before the problem formulation, we first introduce several important notations used in CS-based image compression. The original image signal and its sparse representation basis are denoted by \mathbf{f} and Ψ , respectively. The corresponding sparse signal of \mathbf{f} under Ψ is denoted by \mathbf{s} , namely, $\mathbf{s} = \Psi^{-1}\mathbf{f}$.

Firstly, we review CS-based image compression and decoding for image signal \mathbf{f} . The compression process is expressed as

$$\mathbf{y} = \Phi\mathbf{f} = \Phi\Psi\mathbf{s} \quad (2)$$

where \mathbf{y} presents the measurement vector. The decoding process is expressed as

$$\hat{\mathbf{s}} = \arg \min_{\mathbf{s} \in \mathbb{R}^n} \|\mathbf{s}\|_1 \quad s.t. \quad \mathbf{y} = \Phi\Psi\mathbf{s} \quad (3)$$

where $\hat{\mathbf{s}}$ is the recovery sparse signal corresponding to \mathbf{f} , with $\hat{\mathbf{f}} = \Psi\hat{\mathbf{s}}$. If the attacker does not know Φ and Ψ , the measurement vector \mathbf{y} can be considered as ciphertext [19]. However, if the recovery process is carried out in cloud, the measurement vector \mathbf{y} is no longer private because both Φ and Ψ are public. Then how to outsource expensive CS decoding task to the cloud without revealing the original image signal? Since the measurement matrix has a strong encryption function, can we consider different measurement matrices for encoding and decoding of sampling device and cloud?

Assuming that the decoding measurement matrix used in cloud is Φ , which could be general measurement matrix such as Gaussian random matrix, ± 1 Bernoulli matrix [5]. The encoding measurement matrix is denoted by ΦA , where A is a $n \times n$ matrix. Then, the image compression process is expressed as

$$\mathbf{y}' = \Phi A \mathbf{f} \quad (4)$$

where \mathbf{y}' is measurement vector corresponding to ΦA . Due to $\mathbf{f} = \Psi\mathbf{s}$, Eq. 4 can be expressed as

$$\mathbf{y}' = \Phi A \mathbf{f} = \Phi A \Psi \mathbf{s} \quad (5)$$

If A is invertible, $A\Psi$ can be converted as ΨB , with $B = \Psi^{-1}A\Psi$. Eq. 5 is equivalent to

$$\mathbf{y}' = \Phi A \mathbf{f} = \Phi \Psi B \mathbf{s} = \Phi \Psi \mathbf{s}' \quad (6)$$

where $\mathbf{s}' = B\mathbf{s}$. After image compression, the cloud carries out the decoding operation as

$$\hat{\mathbf{s}}' = \arg \min_{\mathbf{s}' \in \mathbb{R}^n} \|\mathbf{s}'\|_1 \quad s.t. \quad \mathbf{y}' = \Phi \Psi \mathbf{s}' \quad (7)$$

where $\hat{\mathbf{s}}'$ is the recovery signal corresponding to \mathbf{s}' via solving ℓ_1 optimization problem. Therefore, if we can find a suitable matrix A satisfied three conditions: (1) A is invertible. (2) A

can not decrease the sparsity of \mathbf{s} . (3) \mathbf{s} can not be obtained from \mathbf{s}' without A . We can implement secure image data storage and CS decoding in the cloud.

C. Problem Formulation

We next introduce the problem formulation. In the rest of this paper, Φ and A are called measurement matrix and encryption matrix, respectively. If an attacker can not recover \mathbf{f} given \mathbf{s}' and Ψ , we can securely implement image decoding in the cloud. Therefore, our problem is to find an appropriate encryption matrix A^* such that

$$\begin{aligned} (\mathbf{P}) \quad A^* &= \arg \min_{A \in \mathbb{R}^{n \times n}} \mathcal{P}(A, \Phi, \mathbf{f}) \\ s.t. \quad \mathbf{y}' &= \Phi A \mathbf{f} = \Phi \Psi \mathbf{s}' \\ \mathbf{s}' &= \Psi^{-1} A \Psi \mathbf{s} \\ \mathcal{C}(\Phi, A, \mathbf{f}) &\leq \mathcal{C}(\Phi, \mathbf{f}) \end{aligned}$$

where $\mathcal{P}(\cdot)$ is privacy exposure function which will be described later, $\mathcal{C}(\cdot)$ is communication cost function. $\mathcal{C}(\Phi, A, \mathbf{f}) \leq \mathcal{C}(\Phi, \mathbf{f})$ ensures that the transmission cost of our scheme is no larger than the original CS based image compression.

IV. OUR SOLUTION

In this section, we will discuss our encryption matrix design and implement in detail, and give each component design of our eCIS.

A. Encryption Matrix Design and Implement

1) *Design Principle*: Recall that our goal of encryption matrix is to ensure that the original sparse signal of user's image can not be obtained from the recovered sparse signal in the cloud. If the encryption matrix does not change the sparsity of original image signal, we can remove the condition of $\mathcal{C}(\Phi, A, \mathbf{f}) \leq \mathcal{C}(\Phi, \mathbf{f})$ from the problem (P). This is because when the measurement matrix is given, the number of CS measurements which only depends on the sparsity of the compressed signal.

In information theory, mutual information represents the shared information between two random variables according to Definition 1. In this work, we exploit mutual information as our privacy exposure function. Intuitively, it measures what extent \mathbf{s} can be inferred from \mathbf{s}' . We assume that $\mathbf{s} = [s_1, s_2, \dots, s_n]^T$, $\mathbf{s}' = [s'_1, s'_2, \dots, s'_n]^T$, $s_i \in \{a_1, a_2, \dots, a_m\}$, and $s'_i \in \{b_1, b_2, \dots, b_m\}$ for $i = 1, 2, \dots, m$. each of the presentation, we also assume $\xi_A = \{a_1, a_2, \dots, a_m\}$, $\xi_B = \{b_1, b_2, \dots, b_m\}$, $P\{s_k = a_i\} = p(a_i)$, and $P\{s'_k = b_j\} = p(b_j)$. Accordingly, the problem (P) is equivalent to

$$\begin{aligned} (\mathbf{P1}) \quad & \min_{P(\mathbf{s}'/\mathbf{s})} I(\mathbf{s}; \mathbf{s}') \\ s.t. \quad & \mathbf{s}' = B\mathbf{s} \\ & B = \Psi^{-1}A\Psi \end{aligned}$$

According to Definition 1, $I(\mathbf{s}; \mathbf{s}') = H(\mathbf{s}') - H(\mathbf{s}'/\mathbf{s})$. Since \mathbf{s} and \mathbf{s}' can be considered as discrete memoryless n times

extension of single symbol, $H(s')$ is given by

$$\begin{aligned} H(s') &= - \sum_{i_1=1}^n \cdots \sum_{i_n=1}^n p(a_{i_1} \cdots a_{i_n}) \log p(a_{i_1} \cdots a_{i_n}) \\ &= - \sum_{i_1=1}^n \cdots \sum_{i_n=1}^n \prod_{k=1}^n p(a_{i_k}) \log \prod_{k=1}^n p(a_{i_k}) \\ &= nH(\xi_B) \end{aligned}$$

Similarly, $H(s'/s)$ is given by

$$\begin{aligned} H(s'/s) &= - \sum_{i_1=1}^n \cdots \sum_{i_n=1}^n \sum_{j_1=1}^n \cdots \sum_{j_n=1}^n p(a_{i_1} \cdots a_{i_n}) \\ &\quad p(b_{j_1} \cdots b_{j_n}/a_{i_1} \cdots a_{i_n}) \log_2 p(b_{j_1} \cdots b_{j_n}/a_{i_1} \cdots a_{i_n}) \\ &= - \sum_{i_1=1}^n \cdots \sum_{i_n=1}^n \sum_{j_1=1}^n \cdots \sum_{j_n=1}^n p(a_{i_1} \cdots a_{i_n}) \\ &\quad \prod_{k=1}^n p(a_{i_k}) p(b_{j_k}/a_{i_k}) \log \prod_{k=1}^n p(b_{j_k}/a_{i_k}) \\ &= n(H(\xi_B) - H(\xi_B/\xi_A)) \end{aligned}$$

According to Eq.1, $I(s; s')$ is given by

$$\begin{aligned} I(s; s') &= H(s') - H(s'/s) \\ &= n(H(\xi_B) - H(\xi_B/\xi_A)) \end{aligned}$$

In order to minimize $I(s; s')$, it is equivalent to minimizing $H(\xi_B) - H(\xi_B/\xi_A)$. When $H(\xi_B/\xi_A) = H(\xi_B)$, $I(s; s')$ achieves the minimum value ($I(s, s') = 0$), this is because $H(\xi_B)$ and $H(\xi_B/\xi_A)$ are greater than or equal to 0. Note that $H(\xi_B/\xi_A) = H(\xi_B)$ implies that $p(b_j/a_i) = p(b_j)$ for $i, j = 1, 2, \dots, n$. Namely, the events a_i and b_j are independent to each other. In other words, s can not be recovered from s' if all elements are independent to each other.

The goal of encryption matrix is to minimize $I(s; s')$. When the image signal is given, the value of $I(s; s')$ is decided by the conditional probability transform matrix, Γ , which is denoted by

$$\Gamma = \begin{bmatrix} p(b_1/a_1) & p(b_2/a_1) & \cdots & p(b_n/a_1) \\ p(b_1/a_2) & p(b_2/a_2) & \cdots & p(b_n/a_2) \\ \vdots & \vdots & \ddots & \vdots \\ p(b_1/a_n) & p(b_2/a_n) & \cdots & p(b_n/a_n) \end{bmatrix}$$

The conditional probability transform matrix should be implemented by encryption matrix. When encryption matrix make $P(\xi_A)$ and $P(\xi_B)$ independently of each other, our goal that s can not be obtained from s' could be achieved. Although $P(\xi_B)$ can be any probability distribution, we should carefully choose a probability distribution to facilitate our implementation. In this work, we set $P(\xi_B)$ as discrete equivalent probability distribution, namely, $p(b_i) = p(a_j/b_i) = 1/n$ for each i and j . Therefore, Γ can be expressed as

$$\text{Gamma} = \begin{bmatrix} 1/n & 1/n & \cdots & 1/n \\ 1/n & 1/n & \cdots & 1/n \\ \vdots & \vdots & \ddots & \vdots \\ 1/n & 1/n & \cdots & 1/n \end{bmatrix}$$

According to the above $P(\xi_B)$ and Γ , A should implement all the nonzero elements of s with uniform distribution for s' .

2) *Encryption Matrix Implementation*: In this part, we give our encryption matrix A implementation based on above analysis. We exploit random perturbation identity matrix as A to implement the function of conditional probability transformation matrix Γ . The definition of A is

$$A = \pi(I) \quad (8)$$

where I is identity matrix. $\pi(\cdot)$ is random perturbation function with the same probability, which is equivalent to randomly perturbing the rows of I . According to this definition, we know that A is an invertible matrix. Meanwhile, A has encryption function because it could implement all the elements of s and s' to be independent to each other. Since B equals to $\Psi^{-1}A\Psi$ and $s = Bs'$, B only perturbs the element location of s , and the sparsity of s is not changed. So our designed encryption matrix satisfies three conditions of section 3.2. If all the elements of s are perturbed by A , we only need to implement the equal probability random perturbation for each element. In fact, A could be easily implemented by a random seed, which is also considered as the key between the sampling device and the end user.

To quantify the privacy protection level more effectively, and make a more detailed description on security considerations, we next introduce the concept of k -secure.

Definition 2. The encryption matrix A is k -secure if the number of permutation rows is k . It is denoted by

$$A_k = \pi_k(I) \quad (9)$$

A_k is called k -secure.

If $k = 0$, A_0 is equivalent to I , i.e., no encryption operation is carried out. On the other hand, we achieve the highest security level by setting $k = n$, i.e., perturb all elements in original sparse signal. Consider that s is sparse, in order to implement non-zero elements of s' with uniform distribution, we should select more non-zero elements to carry out perturbation operations. If the non-zero elements distribution of s is known, the same probability can be used as perturbation probability for each element. Otherwise, uniform random selection is adopted for each element.

Since A is random perturbation matrix of I , ΦA is equivalent to perturb the column of Φ . The image compression and encryption could be implemented simultaneously.

B. System Design

Encryption and compression component: Sampling device is responsible for sampling, compression and encryption operations. We mainly consider compression and encryption after image signal sampling. In encryption and compression component, we exploit the common measurement matrix, Gaussian random matrix Φ , as our measurement matrix. The encryption and compression process is calculate as $y' = \Phi A f$. Due to A is random perturbation identity matrix by identity

matrix I , it can be implemented by a random seed. ΦA is equivalent to perturb the column of Φ .

Cloud component: In eCIS, cloud component is responsible for storing user's compressed image signal \mathbf{y}' and decoding the encryption sparse signal \mathbf{s}' . If there is no user's request, cloud only stores the user's image signal \mathbf{y}' and Φ . If the user requests an image signal, it sends Ψ to the cloud. The cloud receives the user's request and Ψ , and carries out CS decoding operation according to Eq. 7. After decoding the encryption sparse signal \mathbf{s}' , the cloud sends it to the end user. Considering that cloud could be publicly accessible, the attacker can obtain \mathbf{s}' and Ψ . Although the attacker can obtain the user's data, it can only recover the encryption image signal via $\Psi \mathbf{s}'$ ($A\mathbf{f} = \Psi \mathbf{s}'$). According to our encryption matrix, cloud can implement both storage and decoding computation functions allowing significant reduction of resource consumption on the sampling device and end user.

End user component: The end user is responsible for decrypting and recovering the original image. If the end user receives \mathbf{s}' , it carries out decryption and recovery operations. According to Eq. 6, we know $A\mathbf{f} = \Psi \mathbf{s}'$. Therefore, the recovered signal is given by

$$\mathbf{f} = A^{-1}\Psi \mathbf{s}' \quad (10)$$

Since A is random perturbation identity matrix of identity matrix I , A^{-1} is equivalent to the transpose form of A . Therefore, $\mathbf{f} = A^T \Psi \mathbf{s}'$. $A^T \Psi$ is equivalent to perturbing the rows of Ψ . The image recovery and decryption could be easily implemented, and eCIS does not need complicated decryption operation compared with transformed-based image recovery.

V. THEORETICAL ANALYSIS

A. Security Issue

According to our encryption matrix definition as shown in Eq.9, if the attacker wants to obtain the original sparse signal and A is k -secure, he should investigate all possible arrangements as

$$C(n, k) \times n(n-1) \cdots (n-k+1) = C(n, k) \cdot \frac{n!}{(n-k)!}$$

where $C(n, k) = \frac{n!}{(n-k)!k!}$. If the attacker also does not know the value of k , the number of possible combinations is

$$\sum_{k=1}^n (C(n, k) \cdot \frac{n!}{(n-k)!})$$

Suppose that the attacker knows the value of k , the probability to successfully recover \mathbf{s} , P_{suc} , can be calculated according to Stirling's approximation [12]

$$\begin{aligned} P_{suc} &= \frac{1}{C(n, k) \times \frac{n!}{(n-k)!}} = \frac{1}{C(n, k) \times C(n, k) \times k!} \\ &\leq \frac{1}{e(en^2/k)^k} \leq \frac{1}{e(en)^k} = e^{-(k \log n + k + 1)} \quad (11) \end{aligned}$$

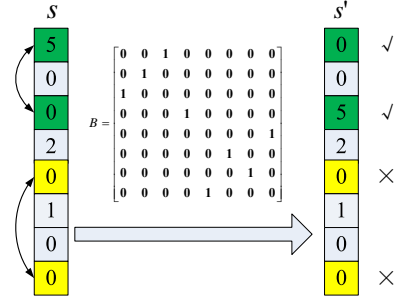


Fig. 2. Example of \mathbf{s} and \mathbf{s}' under A .

If the recovery probability is required to be less than β , k should satisfy the following inequality:

$$k \geq \left\lceil \frac{-\log \beta - 1}{\log n + 1} \right\rceil$$

The signal \mathbf{s} , however, is sparse, the attacker may not need to try too many perturbations of zero elements. To simplify the problem analysis, we do not consider the nonzero element probability distribution of \mathbf{s} . We assume that there are l perturbed elements among \mathbf{s} and \mathbf{s}' to be considered. According to Eq.11, the probability of successfully recovered \mathbf{s} , P_{suc} , is given by

$$P_{suc} = \frac{1}{C(n, l) \times \frac{n!}{(n-l)!}} \leq \frac{1}{e(en)^l} = e^{-(l \log n + l + 1)} \quad (12)$$

Nextly, we calculate the value of l . We suppose that \mathbf{s} is t -sparse and A is k -secure. First of all, we look at what circumstance the attacker does not need to consider. For example, Fig.2 shows that 4 elements are perturbed between \mathbf{s} and \mathbf{s}' , namely, A is 4-secure. The attacker only needs to guess the first two elements 0 and 5, and does not consider the other two perturbed 0 elements. In other words, the attacker does not consider the case of two zero elements perturbation. Since the number of selection zero elements is decided by the non-zero element distribution, we conduct the security analyses with uniform and nonuniform distribution of non-zero elements in \mathbf{s} .

1) *Uniform Distribution:* When the nonzero element follows uniform distribution in \mathbf{s} , each zero element of \mathbf{s} has the same perturbation probability. The probability of two zero elements perturbation is

$$\frac{n-t}{n} \cdot \frac{n-1-t}{n-1} = \frac{(n-t)(n-t-1)}{n(n-1)}$$

Therefore, the average number of zero elements which is not to be considered is

$$k \cdot \frac{(n-t)(n-t-1)}{n(n-1)}$$

The number of perturbed elements which needs to be considered, l , is

$$l = k \cdot \left(1 - \frac{(n-t)(n-t-1)}{n(n-1)}\right) = \frac{kt(2n-t-1)}{n(n-1)}$$

Given that $t(2n-t-1)/(n(n-1))$ is a constant, the number of perturbed elements l can be denoted by $\alpha \cdot k$, where $\alpha = t(2n-t-1)/(n(n-1))$ and $0 < \alpha < 1$. According to Eq.12, P_{suc} can be calculated as

$$P_{suc} \leq \frac{1}{e(en)^{\alpha k}} = e^{-(\alpha k(\log n + 1) + 1)} \quad (13)$$

Eq.13 indicates that the complexity to recover the original signal is $O(n^{\alpha k \log n})$. If the recovery probability is required to be less than β , k should satisfy the following inequality:

$$k \geq \left\lceil \frac{-\log \beta - 1}{\alpha(\log n + 1)} \right\rceil = \left\lceil \frac{n(n-1)(-\log \beta - 1)}{t(2n-t-1)(\log n + 1)} \right\rceil$$

2) *Nonuniform Distribution*: We next study the case when the nonzero element follows nonuniform distribution in \mathbf{s} , we denote the distribution probability is $P(\mathbf{s})$ and the probability of the i^{th} location is $p(s_i)$. We also assume that the element selection probability is $q(s_i)$. The expected probability for choosing an non-zero element in one selection is $\sum_{i=1}^n p(s_i) \cdot q(s_i)$. The probability that we choose two zero elements for a perturbation is $(1 - \sum_{i=1}^n p(s_i)q(s_i))^2$. To reduce the number of selected zero elements, $\sum_{i=1}^n p(s_i)q(s_i)$ should be as large as possible. According to principle of the maximum, $\sum_{i=1}^n p(s_i)q(s_i)$ achieves a maximum value if $q(s_i)$ is equal to $p(s_i)$. In other words, the expected number of selected zero element is minimized if the selection strategy is the same as the nonzero elements distribution.

Since $q(s_i)$ is equivalent to $p(s_i)$, the number of perturbed elements which the attacker needs to consider can be expressed as

$$l = k \cdot (1 - (1 - \sum_{i=1}^n p(s_i)^2)^2)$$

Then, P_{suc} is given by

$$P_{suc} \leq e^{-((1 - (1 - \sum_{i=1}^n p(s_i)^2)^2)k(\log n + 1) + 1)} \quad (14)$$

If the recovery probability is required to be less than β , the value k should satisfy the following inequality

$$k \geq \left\lceil \frac{(-\log \beta - 1)}{((1 - (1 - \sum_{i=1}^n p(s_i)^2)^2) \log n + 1)} \right\rceil$$

When the distribution of nonzero elements is unknown, we consider the perturbed elements as uniform random strategy, and $q(s_i)$ is equivalent to $1/n$. Therefore,

$$\begin{aligned} P_{suc} &\leq e^{-((1 - (1 - \frac{1}{n} \sum_{i=1}^n p(s_i))^2)k(\log n + 1) + 1)} \\ &= e^{-(\frac{2n-1}{n^2}k(\log n + 1) + 1)} \end{aligned} \quad (15)$$

Eq.15 indicates that the complexity to recover the original signal is $O(n^{(k \log n)/n})$. If the recovery probability is required to be less than β , the value of k should satisfy the following inequality.

$$k \geq \left\lceil \frac{(-n^2 \log \beta - 1)}{((2n-1) \log n + 1)} \right\rceil$$

Remark: According to the above security analysis, the user can adjust the security level k according to his privacy requirement. For k -secure requirements, the computational

complexity for attacker is $O(n^{(tk \log n)/n})$ when uniform distribution is concerned for non-zero elements, and $O(n^{(k \log n)/n})$ when non-uniform distribution is concerned.

B. Overhead Analysis

Computational Complexity: In this part, we analyze computational complexity for each component of eCIS. For compression and encryption operations, it calculates $\mathbf{y}' = \Phi \mathbf{A} \mathbf{f}$. Since \mathbf{A} is random perturbation matrix of identity matrix \mathbf{I} , calculating $\Phi \mathbf{A}$ is equivalent to perturbing the column of Φ whose complexity is $O(mn)$. If the sampling device only carries out CS-based compression without encryption, the complexity is also $O(mn)$. If the sampling device takes LP transform outsourcing [25], its complexity is $O(n^\theta + mn)$ ($2 < \theta < 3$).

For end user component, the decryption and recovery processes are $\mathbf{f} = \mathbf{A}^{-1} \Psi \mathbf{s}'$, and \mathbf{A}^{-1} is equivalent to \mathbf{A}^{-T} . Computing $\mathbf{A}^T \Psi$ only requires to perturb the rows of Ψ , thus the complexity of end user component is $O(n^2)$. Assume the end user computes the CS solution via Eq. 3 without outsourcing it to the cloud, the computation cost would be $O(n^3)$ (e.g [23]).

At the cloud side, the computation cost is $O(n^3)$ which is equivalent to the complexity of solving Eq. 7. In eCIS, we shift the ℓ_1 optimization from the end user to the cloud without adding extra computation cost. More importantly, eCIS provides privacy protection compared with the original cloud-assisted CS decoding.

Communication Cost: In eCIS, the transmission cost between the sampling device and cloud is measured by the number of CS measurements that are transmitted. Since our scheme does not change the sparsity of the original image signal, the transmission cost remains the same as that of the nonencryption CS-based image data compression. Although the scheme proposed in [25] has the same transmission cost between the sampling device and cloud, the transmission cost between cloud and end user is very high because the cloud needs to send the encrypted original image data to the end user. In our scheme, cloud only sends the encrypted sparse signal to the end user allowing great reduction of the transmission cost. For example, when the compression ratio is no less than 50%, our scheme can reduce half of the transmission cost compared with the scheme proposed in [25].

VI. EXPERIMENT

In this section, we carry out extensive experiments to evaluate the performance of eCIS. In our experiment, we use different test sequences as the signal source of resource-constrained device. Our experiment is implemented via MATLAB, on a laptop with an Intel Core i5 CPU running at 1.6GHz and 4G RAM. Gaussian random matrix is considered as our measurement matrix. Since the original image signal is not sparse in spatial domain, we select DCT basis as sparse representation basis because it is the most common sparse representation basis for image compression. In order to efficiently implement eCIS, the experimental images are divided

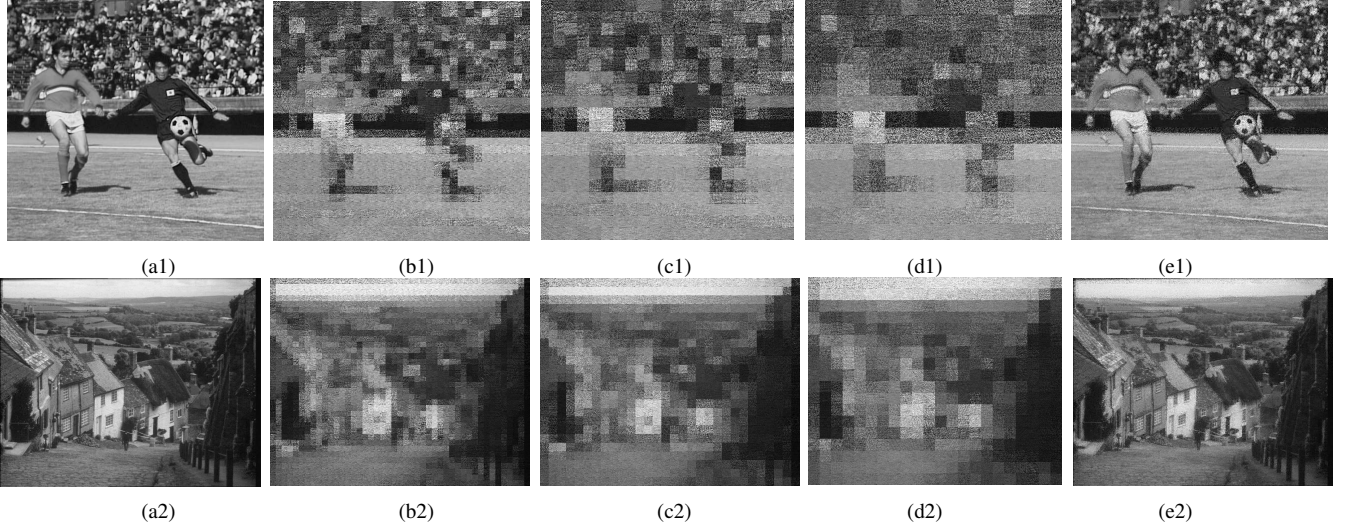


Fig. 3. Cloud-assisted image encryption with different block sizes. (a1) and (a2). Original images. (b1) and (b2). 16×16 pixels of block size. (c1) and (c2). 24×24 pixels of block size. (d1) and (d2). 32×32 pixels of block size. (e1) and (e2). Recovery image of end user.

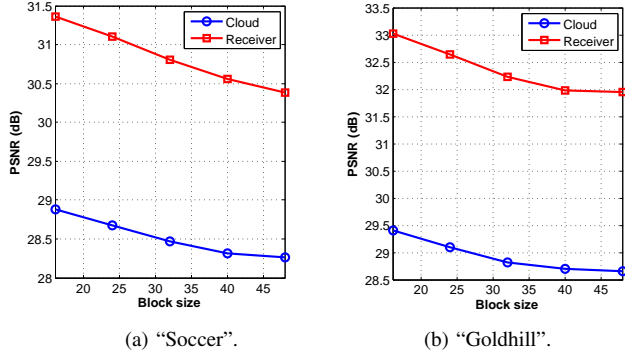


Fig. 4. PSNR comparison of recovery image between cloud and end user with different block size.

into multiple the same size blocks. We also transform 2-dimension block pixel values into 1-dimension signal via row sequence to meet CS requirement. Meanwhile, we evaluate the performances of our scheme from two aspects: effectiveness and computation overhead.

A. Effectiveness Evaluation

Our goal of effectiveness evaluation is to display image recovery performance of cloud and end user. We evaluate its performance from three aspects: different block size, different security level, and adaptive region of interest (ROI) of test images. In our experiment, subjective visual effect and peak signal-to-noise ratio (PSNR) are considered as metrics to evaluate the quality of the recovery image. PSNR is defined as

$$PSNR = 10 \times \log_{10} \frac{255^2}{MSE}$$

where MSE is mean square error of gray scale pixel values between the original and recovery images.

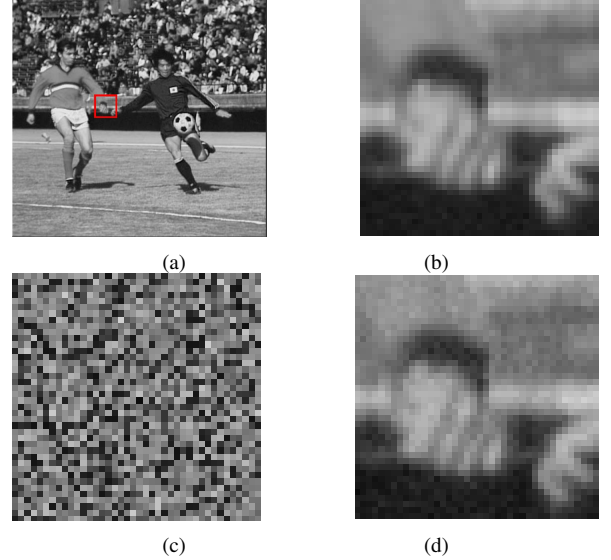


Fig. 5. Cloud-assisted image encryption of one block. (a) Original image and selected block. (b) 40×40 pixels block. (c) Cloud recovery image with n -secure. (d) Recovery image of end user.

1) *Impact of Block Size*: In our experiment, the size of the division image block needs to consider the following three issues:

1) The block size affects the secure performance of eCIS. The greater the size of division block is, the smaller the probability that the original signal is successfully recovered by the attacker.

2) The block size affects the scale of ℓ_1 optimization problem, namely, Eq.7. The greater size Eq.7 solves, the more time the problem requires.

3) The block size affects the compression performance of image source and the communication cost between sampling

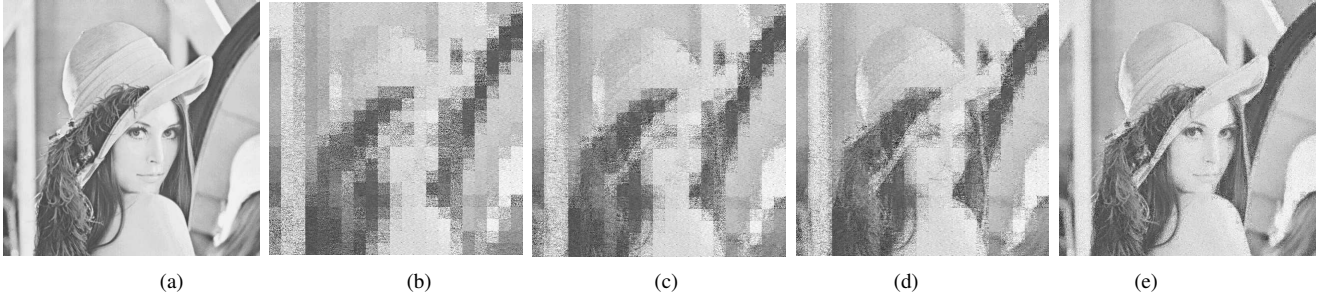


Fig. 6. Cloud-assisted image encryption with different security level. (a) Original image of “Lenna”. (b) n -secure. (c) $n/2$ -secure. (d) $n/3$ -secure. (e) Recovery image of end user.

device and cloud. The greater the size of division block is, the better the sparsity of compression block cloud obtain. The sparse level decides the number of transmission CS measurements, which affects the transmission cost from sampling device to cloud.

Fig.3 displays the experiment results of our scheme with different block sizes and n -secure with uniform distribution random perturbation. Fig.3 (a1) and (a2) are the original image sources of “Soccer” and “Goldhill”. Fig.3 (b1) and (b2) display the cloud recovery image with 16×16 pixels of block size. Fig.3 (c1) and (c2) are 24×24 pixels of block size. Fig.3 (d1) and (d2) are 32×32 pixels of block size. Fig.3 (e1) and (e2) are recovery image of end user. According to Fig.3, it illustrates that the recovery images of cloud become more ambiguous with the increasing the block size. Fig.4 displays PSNR comparison of recovery image between cloud and end user across different block size. The experiment is implemented under the same number of measurements for the same block size. It shows that the recovery numerical performance of end user is much better than the cloud. In cloud, the PSNR of “Soccer” and “Goldhill” are less than 29dB and 29.5dB, respectively.

Fig.5 displays experimental result of cloud-assisted image encryption with 40×40 pixels of block size of image “soccer”. The red image block of Fig.5 (a) is our selected block. The experiment was implemented with n -secure and uniform distribution random perturbation. The experimental results displays that the image details are invisible at all as shown in Fig.5 (c). However, each encryption block still represents average pixel values of the original image from visual aspect as shown in Fig.3. The reason is that the mainly energy of block under DCT basis focus on only one few low frequency coefficients. If the user wants to make the encryption image become more ambiguous from visual aspect, we could increase the division block size.

2) *Security Level Considerations*: In this part, we demonstrate the experiment results of eCIS with different security levels. Fig.6 shows that our results on image “Lenna” and under uniform distribution random perturbation. Fig.6 (a) displays the original image. Fig.6 (b), (c), and (d) display the recovered images using cloud sparse signal with n , $n/2$ and $n/3$ -secure, respectively. Fig.6 (e) displays the recovered

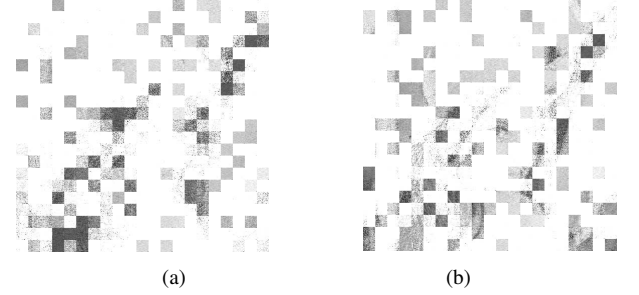


Fig. 7. Cloud-assisted image encryption with uniform random perturbation and amplitude random encryption. (a) $n/2$ -secure. (b) $n/3$ -secure.

image at the end user. According to Fig.6, the recovered image using cloud sparse signal leaks more details about the original image with increasing the security level. The attacker can obtain almost contour and many details of original image when the encryption matrix is $n/3$ -secure as shown in Fig.6 (d).

In order to enhance the visual encryption effectiveness, we could further encrypt the amplitude of sparse signal with random multiplication. For example, we can adopt the encryption process as described in Eq.16 for each image block.

$$A_k = \alpha \cdot \pi_k(I) \quad (16)$$

where α is also a random value. It means that the encryption pixel values of each block are multiplied by a same random value. Fig.7 shows the experiment results of “Lenna” image with $n/2$ and $n/3$ -secure. The value of α is random selected between 0 and 1. The experiment result displays that we only obtain a little image information from visual aspect. However, it cannot improve much help from attack aspect because all the pixel values are carried out the same linear operation. The goal of random amplitude encryption only perturbs the image contour from visual aspect.

3) *ROI Encryption*: eCIS can also perform encryption based on the user’s region of interest requirement. For example, Fig.8 displays the experiment result of our scheme with ROI image encryption and 24×24 pixels of block size. Fig.8 (a) is the original “Barbara” image, the red block is the encryption block of ROI. Fig.8 (b), (c) and (d) are the recovered images according to the cloud recovery sparse signal. Fig.8

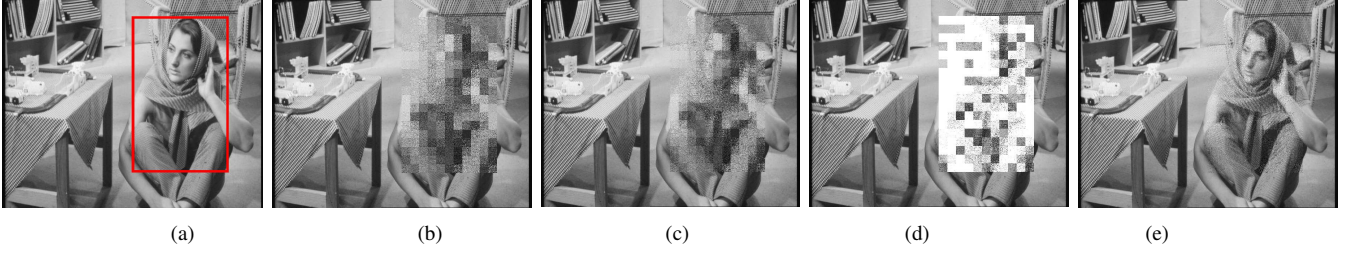


Fig. 8. Cloud-assisted image encryption with region of interest. (a) Original image and selected ROI of “Barbara”. (b) n -secure. (c) $n/2$ -secure. (d) $n/2$ -secure with random amplitude encryption. (e) Recovery image of end user.

(b) and (c) are n -secure and $n/2$ -secure, respectively. Fig.8 (d) is the $n/2$ -secure with random amplitude encryption and $0 < \alpha < 1$. Fig.8 (e) is the recovery image of end user. The experiment result displays that we almost can not know the details and contour of selected ROI according to Fig.8 (b) and (d). Fig.8 (d) is sufficient ambiguous from visual aspect.

B. Overhead Evaluation

In this subsection, we evaluate the overhead of eCIS. We mainly focus on computation cost at the sampling device and end user. In order to effectively evaluate the overhead of our scheme, we compare eCIS with the following existing schemes: (1) CS-based image process without cloud service and security consideration (Original_CS), (2) Cloud-assisted and CS-based image process without security consideration (Cloud_Non_encryption), and (3) Cloud-assisted and CS-based image process with security consideration by LP problem transformation [25].

For ease of presentation, we implement different schemes with the same size of image block, 24×24 , 32×32 , and 48×48 pixels of block size. All the results represent the mean of 50 experiments and each experiment is randomly selected one block from the same test image. We implement our scheme with uniformly random perturbation and n -secure. OMP algorithm is selected to solve ℓ_1 optimization problem [23]. Let t_{sd} and t_{eu} denote the running time of sampling device and end user, respectively. In this experiment, we do not consider the image sampling time. t_{sd} is the image compression time for Original_CS and Cloud_Non_encryption schemes. For our scheme, t_{sd} is the compression and encryption time. t_{eu} is the time of solving ℓ_1 optimization problem and image recovery for Original_CS scheme. t_{eu} is only the time of image recovery for Cloud_Non_encryption schemes. In our scheme, t_{eu} is the time of image decryption and recovery.

Table.I displays the mean running time (in seconds) comparisons of Original_CS, Cloud_Non_encryption, and eCIS. The last column of Table.I represents the system speedup compared our scheme with Original_CS scheme. The first and the second sub-columns represent the time speedup of sampling device and end user, respectively. The last sub-column is the total time ($T_{total} = T_{sd} + T_{eu}$) speedup without considering the time spent on the cloud. According to Table.I, our running time only increases $1.2 \times \sim 1.7 \times$ to obtain encryption function for sampling device under different

block sizes compared with Original_CS scheme. In [25], LP problem transformation scheme requires $5.6 \times$ and $9.4 \times$ time cost compared with Original_CS scheme to obtain encryption function for 32×32 , 48×48 pixels of block sizes, respectively. Meanwhile, our scheme can decrease $4.6 \times \sim 7.7 \times$ time cost for end user compared with Original_CS scheme. For total time speedup, the experimental result displays that our eCIS decrease $4.1 \times \sim 6.8 \times$ time cost. Even when compared with Cloud_Non_encryption schemes, our running time only increases by up to $2 \times$ in end user. In [25], the authors didn't consider the final image recovery cost. They decreases the total running time $4.0 \times$ and $3.4 \times$ with 32×32 and 48×48 pixels of block size, respectively. Without considering the image recovery time, our scheme decreases the total running time $8.9 \times$ and $11.7 \times$ with 32×32 and 48×48 pixels of block size of “Lenna”, respectively. For the image of “Soccer”, our scheme decrease the total running time by $8.37 \times$ and $12.7 \times$ with 32×32 and 48×48 pixels of block sizes, respectively. The experiment result shows that eCIS can keep the user's image privacy with low-complexity compared with the existing cloud-assisted image service scheme.

VII. RELATED WORK

Image compression and encryption: Image compression technology can be divided into two categories, transform-based compression and CS-based compression. Existing image encryption technology is mainly aimed at transform-based compression image such as [8], [17], [26]. Transform-based compression technology requires high computation complexity and storage cost for encoder. This type of image compression technology is not suitable for resource-constrained smart device. CS-based image compression technology can shift the complexity from encoder to decoder. Although CS theory and its applications have received lots of researches in recent years [7], [18], [9], [29], [15], CS-based image encryption technology also require high computation complexity [16], [28]. These methods are not appropriate for resource-constrained smart device. For resource-constrained smart device user, these two types of compression and encryption technology not directly meet the requirement of user's privacy and system resource.

Secure computation outsourcing: With the development of cloud computing in recent years, cloud provides an new avenue for storage and computation outsourcing according to

TABLE I
MEAN RUNNING TIME COMPARISON OF DIFFERENT BLOCK SIZE (IN SECONDS).

Image	Block size	Original_CS		Cloud_Non_encryption		Our scheme		Speedup		
		T_{sd}	T_{eu}	T_{sd}	T_{eu}	T_{sd}	T_{rec}	T_{sd}	T_{eu}	T_{total}
Lenna	24×24	0.0056	0.17824	0.00566	0.0263	0.00904	0.0344	$-1.7 \times$	$5.2 \times$	$4.2 \times$
	32×32	0.01828	0.64044	0.0168	0.0835	0.02278	0.13748	$-1.3 \times$	$4.6 \times$	$4.1 \times$
	48×48	0.08816	5.73816	0.0884	0.40624	0.1125	0.79696	$-1.3 \times$	$7.2 \times$	$6.4 \times$
Soccer	24×24	0.0056	0.16688	0.00498	0.02654	0.00694	0.03284	$-1.2 \times$	$5.1 \times$	$4.3 \times$
	32×32	0.0175	0.65472	0.01784	0.08304	0.02808	0.14042	$-1.6 \times$	$4.7 \times$	$4.0 \times$
	48×48	0.08764	5.8582	0.09118	0.41128	0.11528	0.76554	$-1.3 \times$	$7.7 \times$	$6.8 \times$

the user's demand [2]. One advantage of cloud paradigm is resource outsourcing for resource-constrained smart device. However, cloud is public and exists lots of security threat such as data transmission security, data storage security and so on [27], [21]. Cloud-based computation outsourcing has also been studied by many researchers. For example, C. Wang *et. al* have proposed a secure and practical outsourcing of linear programming scheme in cloud computing [24]. In [13], [3], the authors also proposed many schemes to encrypt input and output and implement secure computation outsourcing. These methods are all following the thought of fully homomorphic encryption [14], and transform the original LP problem into another LP problem. Since CS decoding is equivalent to solve LP problem, C. Wang *et.al.* proposed a cloud-assisted computation outsourcing scheme for healthcare video monitoring [25]. This proposal also require high complexity transformation operation to implement secure computation outsourcing.

VIII. CONCLUSION AND FUTURE WORK

In this paper, we discussed a novel encrypressive low-complexity cloud-assisted image service scheme via compressive sensing. Although traditional image encryption and secure linear programming outsourcing techniques can efficiently protect image privacy, these methods are not appropriate for resource-constrained device because of high computational cost. Our scheme can efficiently transform the computation and storage cost to the cloud without increasing transmission cost, and protect image privacy according to user's adaptive security demand. Extensive experiment results demonstrated our scheme can significantly save the system running time. In our scheme, we only consider one-dimension CS-based image compression and encryption. Future work will extend to two-dimension image compression and encryption to further reduce network transmission cost.

REFERENCES

- [1] Hande Alemdar and Cem Ersoy. Wireless sensor networks for healthcare: A survey. *Computer Networks*, 54(15):2688–2710, 2010.
- [2] Michael Armbrust, Armando Fox, Rean Griffith, Anthony D Joseph, Randy Katz, Andy Konwinski, Gunho Lee, David Patterson, Ariel Rabkin, Ion Stoica, et al. A view of cloud computing. *Communications of the ACM*, 53(4):50–58, 2010.
- [3] Mikhail J Atallah and Keith B Frikken. Securely outsourcing linear algebra computations. In *Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security*, pages 48–59. ACM, 2010.
- [4] Richard Baraniuk. Compressive sensing. *IEEE signal processing magazine*, 24(4), 2007.
- [5] E.J. Candès, J. Romberg, and T. Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *Information Theory, IEEE Transactions on*, 52(2):489–509, 2006.
- [6] E.J. Candès and T. Tao. Decoding by linear programming. *Information Theory, IEEE Transactions on*, 51(12):4203–4215, 2005.
- [7] Recht B. Candès E J. Exact matrix completion via convex optimization. *Foundations of Computational mathematics*, 9(6):717–772., 2009.
- [8] Wen Chen, Xudong Chen, and Colin JR Sheppard. Optical image encryption based on diffractive imaging. *Optics letters*, 35(22):3817–3819, 2010.
- [9] Yi-Chao Chen, Lili Qiu, Yin Zhang, Guangtao Xue, and Zhenxian Hu. Robust network compressive sensing. In *Proceedings of the 20th annual international conference on Mobile computing and networking*, pages 545–556. ACM, 2014.
- [10] Ronald A. DeVore. Nonlinear approximation. *Acta numerica*, (7):51–150, 1998.
- [11] David L Donoho. Compressed sensing. *Information Theory, IEEE Transactions on*, 52(4):1289–1306, 2006.
- [12] William Feller. An introduction to probability theory and its applications. *John Wiley & Sons*, 1:50–53, 1968.
- [13] Rosario Gennaro, Craig Gentry, and Bryan Parno. Non-interactive verifiable computing: Outsourcing computation to untrusted workers. In *Advances in Cryptology CRYPTO 2010*, pages 465–482. Springer, 2010.
- [14] Craig Gentry. Fully homomorphic encryption using ideal lattices. In *STOC*, volume 9, pages 169–178, 2009.
- [15] Xinbing Wang X. Tian H. Zheng, S. Xiao. Energy and latency analysis for in-network computation with compressive sensing in wireless sensor networks. In *INFOCOM, 2012 Proceedings IEEE*, pages 2811 – 2815, 2012.
- [16] Rong Huang and Kouichi Sakurai. A robust and compression-combined digital image encryption method based on compressive sensing. In *Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP), 2011 Seventh International Conference on*, pages 105–108. IEEE, 2011.
- [17] Zhengjun Liu, Lie Xu, Chuang Lin, Jingmin Dai, and Shutian Liu. Image encryption scheme by using iterative random phase encoding in gyrator transform domains. *Optics and Lasers in Engineering*, 49(4):542–546, 2011.
- [18] D. Needell and J.A. Tropp. Cosamp: Iterative signal recovery from incomplete and inaccurate samples. *Applied and Computational Harmonic Analysis*, 26(3):301–321, 2009.
- [19] Adem Orsdemir, H Oktay Altun, Gaurav Sharma, and Mark F Bocko. On the security and robustness of encryption via compressed sensing. In *Military Communications Conference, 2008. MILCOM 2008. IEEE*, pages 1–7. IEEE, 2008.
- [20] Mohammed Shoaib and Harinath Garudadri. Digital pacer detection in diagnostic grade ecg. In *e-Health Networking Applications and Services (Healthcom), 2011 13th IEEE International Conference on*, pages 326–331. IEEE, 2011.
- [21] Subashini Subashini and V Kavitha. A survey on security issues in service delivery models of cloud computing. *Journal of Network and Computer Applications*, 34(1):1–11, 2011.
- [22] Thomas M. Cover Joy A. Thomas. *Elements of information theory*. Wiley Series in Telecommunications and Signal Processing, pages 20–21, 2006.

- [23] J.A. Tropp and A.C. Gilbert. Signal recovery from random measurements via orthogonal matching pursuit. *Information Theory, IEEE Transactions on*, 53(12):4655–4666, 2007.
- [24] Cong Wang, Kui Ren, and Jia Wang. Secure and practical outsourcing of linear programming in cloud computing. In *INFOCOM, 2011 Proceedings IEEE*, pages 820–828. IEEE, 2011.
- [25] Cong Wang, Bingsheng Zhang, Kui Ren, Janet M Roveda, Chang Wen Chen, and Zhen Xu. A privacy-aware cloud-assisted healthcare monitoring system via compressive sensing. 2014.
- [26] Yong Wang, Kwok-Wo Wong, Xiaofeng Liao, and Guanrong Chen. A new chaos-based fast image encryption algorithm. *Applied soft computing*, 11(1):514–522, 2011.
- [27] Qi Zhang, Lu Cheng, and Raouf Boutaba. Cloud computing: state-of-the-art and research challenges. *Journal of internet services and applications*, 1(1):7–18, 2010.
- [28] Xinpeng Zhang, Yanli Ren, Guorui Feng, and Zhenxing Qian. Compressing encrypted image using compressive sensing. In *Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP), 2011 Seventh International Conference on*, pages 222–225. IEEE, 2011.
- [29] Yin Zhang, Matthew Roughan, Walter Willinger, and Lili Qiu. Spatio-temporal compressive sensing and internet traffic matrices. In *ACM SIGCOMM Computer Communication Review*, volume 39, pages 267–278. ACM, 2009.